Docket No.: 95-508 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of	:	
	:	
ACHARYA	:	
	:	
Serial No.: 09/905,080	:	Group Art Unit: 2155
	:	
Filed: July 16, 2001	:	Examiner: NGUYEN, Thu Ha T

For: ARRANGEMENT FOR REDUCING APPLICATION EXECUTION BASED ON A DETERMINED LACK OF FLOW CONTROL CREDITS FOR A NETWORK CHANNEL

MAIL STOP: APPEAL BRIEF – PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY BRIEF

Sir:

This Reply Brief is submitted under 37 C.F.R. §41.41 in response to the Examiner's Answer mailed July 17, 2006 and July 25, 2006 (the July 25, 2006 Answer is identical to the July 17, 2006 Answer). This Reply Brief is submitted within two (2) months from the July 17, 2006 date of the Examiner's Answer.

As demonstrated below, the Examiner's Answer is replete with inconsistencies in the rejection under §102 and mischaracterizations of the applied reference (U.S. Patent Pub. No. 2002/0085493 by Pekkala et al): these inconsistencies and mischaracterizations demonstrate the Examiner has failed to establish that Pekkala et al discloses each and every element *as arranged in the claims under review*, as required under 35 USC §102.[1]

[1]See footnotes 2, 4, 8, and 9 on pages 12, 15, and 19 of the Appeal Brief filed April 19, 2006.

A1. Response to Examiner's Comments Regarding Argument A1

As argued on pages 6-11 of the April 19, 2006 Appeal Brief, each of the independent claims 1 and 7 explicitly require that the ***network interface*** outputs "a data flow interruption request based on the detected depletion of flow control resources." The Examiner in the Answer argued on pages 3, 9 and 10, that Pekkala et al discloses the claimed "network interface" in the form of the disclosed port 208 or 608[2], (note the Examiner fails to positively identify the claimed "network interface" in Pekkala et al with respect to claim 7): Pekkala et al. in fact discloses that the switch 106 of Figs. 1, 2 and 6 includes IB ports 208 in Fig. 2, and IB ports 608 in Figs. 6 and 7 that interface with a link partner (752 of Fig. 7) via an IB serial link 132 (para. 44).

However, since the Examiner relies on the IB ports 208 or 608 as a disclosure of the claimed "network interface", the rejection is legally inadequate because the Examiner is unable to demonstrate that the disclosed IB ports 208 or 608 perform the claimed "outputting a data flow interruption request", as required by claims 1 and 7. The Examiner has failed to rebut Appellant's argument at pages 6-11 that the broadest reasonable interpretation of the claimed "data flow interruption request" cannot be so broadly construed as to encompass the disclosed "flow control packet" (output by the IB ports 208 or 608) between distinct network nodes; hence, the disclosed IB ports 208 or 608 do not perform the claimed "outputting a data flow interruption request", as claimed.

In fact, the Examiner does not dispute that the broadest reasonable interpretation of the claimed "data flow interruption request" must be distinct from the flow control information (e.g., flow control packet) sent between the claimed "network node" and a link partner (argued on pages 6-11); rather, the Examiner apparently concedes Applicant's argument because the Examiner no longer argues that the disclosed flow control packet is a disclosure of the claimed "data flow interruption request".

Rather, the Examiner attempts to salvage the rejection by *changing* the element that

---

[2]The Answer argues the port 208 of Figs. 1, 2 as a teaching of the claimed network interface at page 3, and the port 608 of Figs. 6 and 7 as a teaching of the claimed network interface at page 9, lines 11-12 (sec. A3.a.) , and page 10, line 3.

outputs the claimed "data flow interruption request" from the network interface (as required in the claims), to a device that is not a network interface, namely the **buffer control logic 606**:

> after detecting there is no buffer available to receive packet [sic] or there is not enough flow control credits to transmit the packet, the buffer control logic 606 send [sic] notification via signal 744 to flow control logic 726 to shutdown the link partner 752[.]
>
> (Answer at pages 7-8).

Hence, it appears the Examiner now argues that the signal 744 is a disclosure of the claimed "data flow interruption request".

However, the signal 744 is not output by a "network interface", as required by the claims, but a **buffer control logic 744**: this buffer control logic 744 of Fig. 7 cannot be the claimed network interface because it is part of the transaction switch 602, which is ***distinct*** from the IB port 608 of Fig. 7 (see para. 80, 83, and 110-112 of Pekkala et al); further, the buffer control logic 744 is configured for outputting the signal 744 in response to a determined depletion of buffer memory 604 in order to *cause* the flow control logic 726 to output a flow control packet advertising **reduced** flow control credits (i.e., a depletion of flow control resources) (para 112-113). Hence, the buffer control logic 744 does not detect a depletion of flow control resources *representing a depletion of network bandwidth for a prescribed data stream*, as required by the claims, but detects a depletion of buffer memory in the shared buffers 604 and in response **causes** the depletion of flow control resources by outputting the signal 744 to the flow control logic 726.

Finally, the buffer control logic 744 is *also* relied on by the Examiner as a disclosure of the claimed ***memory controller*** of claims 3, 8 and 13.

Hence, the Examiner is applying a piecemeal application of the reference in attempting to synthesize an ill-founded rejection by: (1) relying on the IB ports 208 or 608 as the claimed "network interface" (except that the IB ports 208 and 608 do not output the claimed "data flow interruption request", as required); (2) relying on the buffer control logic 744 as the claimed device that outputs the claimed "data flow interruption request" (except that the buffer control logic is not a "network interface", as claimed because it does not detect "a depletion of flow control resources representing a depletion of network bandwidth for a prescribed data stream", as claimed); and (3) relying on the buffer control logic 744 *also* as a disclosure of the claimed "memory controller" (which also fails for reasons discussed *infra*).

As apparent from the foregoing, the Answer fails to even demonstrate that Pekkala et al discloses the claimed network interface *as arranged in the claims,* namely (1) configured for detecting a depletion of flow control resources representing a depletion of network bandwidth for a prescribed data stream, ***and*** (2) outputting a data flow interruption request ***based on the detected depletion*** of flow control resources. For this reason alone the rejection should be reversed.

The Examiner's above-quoted argument from pages 7-8 that of the buffer control logic 606 outputting "signal 744 to flow control logic 726 to shutdown the link partner 752" also disregards the claimed limitation of reducing the prescribed data stream that is generated ***in the network node***. As described in further detail below, the claims specify reducing, ***by a processor in the network node***, the prescribed data stream by reducing execution of the prescribed ***application resource*** that ***generates*** the data stream. Therefore, the claimed data flow interruption request is used to ***reduce generation*** of the data stream ***in the network node***, and <u>not</u> in the link partner.

A2 Response to Examiner's Comments Regarding Argument A2

As argued on pages 12-15 of the Appeal Brief, independent claims 1 and 7 each require that the processor reduces the prescribed data stream by ***reducing execution of a prescribed application resource, configured for <u>generating</u> the prescribed data stream***, based on the data flow interruption request (already conceded by the Examiner to be distinct from the flow control signal). The prescribed application resource is configured for generating <u>not</u> just a unitary data packet, but a <u>data stream</u> (see, e.g., page 4, lines 22-26 of the specification).

The Examiner on pages 8-9 of the Answer also does not dispute that the broadest reasonable interpretation of "reducing execution of a prescribed application resource" requires that actual generation of the data stream be at least reduced, based on reducing execution of the prescribed application resource that generates the data stream. Rather, the Examiner asserts that a *link partner* is "shut down" to prevent from "transmitting [the] packet 300".

The Examiner's argument, however, ignores the very definition of the term "shut down" as applied in the reference, and belies the fact that the"shut down" of the link partner does *not* provide a disclosure of ***reducing the execution*** of the application generating the data stream, but (as admitted by the Examiner) preventing the **<u>transmission</u>** of data packets *from* the link partner that have already been generated (created) *by* the link partner:

> Over-advertising is possible because ***the IB port can transmit flow control packets to completely stop the link partner from <u>transmitting</u> data packets*** in much less time than the link partner can transmit the over-advertised amount of packet data. That is, ***the port can shut down the link partner well before the link partner can consume the over-advertised credits***, thereby avoiding packet loss due to buffer overrun. *The port shuts down the link partner by advertising zero credits for each VL to the link partner*, as will be described below in detail.
>
> (Para. 74, lines 13-23).

The Examiner's own argument in the last paragraph of page 8 is self-contradictory, because it states on the one hand that the flow control logic 726 shuts down the link partner 752 "to prevent from transmitting packet 300", while on the other hand "the packet 300 [sic] is placed into the inline spill buffer 612 and waited [sic] for the buffer to become free/available to transmit the packet [sic]".

In the former case of the flow control logic 726 being used "to shut down the link partner 752 to prevent from ***transmitting*** packet 300", the Examiner's own statement demonstrates that the intent is to prevent <u>transmission</u> of **<u>the packet</u>**, based on the valid premise that **<u>the packet has already been generated</u>**! In other words, if the packet was <u>not</u> generated, there would be no concern for transmission of "the packet" because the packet does not yet exist. Consequently, this first case in the Examiner's argument fails to demonstrate that Pekkala et al disclose ***reducing execution of the prescribed application resource*** configured for ***generating*** the prescribed <u>data stream</u>.

In the latter case of "the packet 300 [sic] is placed into the inline spill buffer 612 and waited [sic] for the buffer to become free/available to transmit the packet [sic]," this simply cannot be considered a disclosure of "reducing execution" as claimed because the storage in memory of the packet concedes the fact that **the packet has already been generated**. Pekkala et al. explicitly specifies that any packet 300 placed "into the inline spill buffer 612" refers to a packet that is **received from an IB port 608**: "[a]n inline spill buffer 612 **receives packet 300 from its corresponding port 608**, independent of the VL 614 specified, and selectively provides the data to an available shared buffer 604 or stores the data until a shared buffer 604 becomes available to store the data" (para. 77, lines 2-7; see, e.g., para. 76-77).

The only consistency in the Examiner's argument is the initial premise that requires that the data packet already has been generated by the link partner: in the former case the packet has not been transmitted by the link partner due to flow control, and in the latter case the data packet has been transmitted by the link partner, received by the IB port 608, and stored in the inline spill buffer 612 for *subsequent retransmission on an output port.* As apparent from the foregoing, the premise that the data packet has already been generated contradicts the claimed reducing execution of the prescribed application feature configured for ***generating the prescribed data stream***.

Finally, the Examiner's tortured arguments disregard the simple fact that **the switch 106 / 700 of Pekkala et al does not *generate* data streams, but *transfers* data streams by *forwarding received packets*!** Therefore, the switch of Pekkala et al is incapable of reducing execution of the prescribed application resource in the network node because the switch does not generate any data streams, as claimed, but **transfers** data packets between end nodes (e.g., end units 102, 108, see para. 44-45).

Hence, Pekkala et al. fails to disclose or suggest the claimed reducing execution of a prescribed application resource configured for generating the prescribed data stream, and any assertions by the Examiner to the contrary are insufficient to overcome the deficiencies in the applied reference. "A prior art patent is a reference only for that which it teaches." *Corning Glass v. Sumitomo Electric*, 9 USPQ2d 1962, 1970 (Fed. Cir. 1989).

For these and other reasons, the Answer has failed to establish that Pekkala et al. described each and every feature of the independent claims *in the manner claimed*; hence, the

§102 rejection must be reversed.

A3. Response to Examiner's Comments Regarding Argument A3

As argued in the Appeal Brief:

> Each of the independent claims 1 and 7 specify reducing a prescribed data stream <u>in a network node</u> by reducing execution of a prescribed application resource configured for generating the prescribed data stream. In particular, each independent claim specifies that <u>each</u> of the features of detecting a depletion of flow control resources, outputting a data flow interruption request based on the detected depletion of flow control resources, and reducing execution of the prescribed application resource that generates the prescribed data stream, ***are performed in the same network node***.

(Appeal Brief at 15-16, emphasis in original).

In fact, claim 1 specifies "reducing, by a processor in the network node and based on the data flow interruption request, the prescribed data stream by ***reducing execution*** of a prescribed application resource configured for generating the prescribed data stream", and claim 7 specifies "a processor configured for ***executing a prescribed application resource for generating of the prescribed data stream*** [and] reducing the prescribed data stream by reducing execution of the prescribed application resource". In other words, the reducing execution includes reducing the actual generation of the data stream *in the network node.*

The Examiner does not dispute Appellant's argument of pages 15-19 that the broadest reasonable interpretation requires *all* of the claimed operations are performed in the same network node, but concedes Appellent's argument on page 9 of the Answer by asserting that Pekkala et al. "does teach detecting, outputting, and reducing step within the IB ([InfiniBand™]) switch (i.e., within network node) 106 or 700 of [F]igures 1, 6, and 7."

The Examiner also asserts that "it is inherent that the IB switch has to have a processor **<u>in order [to] execute and control all of control unit, other logics and buffers/memory</u>**." This assertion of inherency, however, does not address the claimed feature of "the processor configured for ***executing*** a prescribed application resource ***for generation of the prescribed data stream***" in the network node, as claimed. Rather, any processor in the IB switch would be to manage **switching operations** such as buffer management, forwarding **received** packets, and flow control: as the Examiner has earlier admitted, the flow control operations are performed to halt transmission in **the link partner 752**, and not the in the switch 106 or 700 (which is asserted as the disclosure of the "same network node").

Hence, the Examiner cannot reconcile the inconsistency between conceding that the claims require all operations (including generation of the prescribed data stream) be performed in the same network node, and the Examiner's prior assertion that halting transmission of a packet already generated by a link partner should be a disclosure of reducing the prescribed data stream by reducing execution of the prescribed application resource *in the network node* that generates the prescribed data stream.

For these and other reasons, the §102 rejection must be withdrawn.

B. Response to Examiner's Comments Regarding Argument B

The Examiner's comments regarding Argument B demonstrate a remarkable disregard for the explicit claim limitations as specified on pages 20-22 of the Appeal Brief. The Examiner simply cites the buffer control logic 606 as the claimed memory controller.

However, the Examiner fails to address the explicit limitations that the memory controller renders unavailable the system memory resources in response to ***reception of the data flow interruption request***. As argued supra, the disclosed buffer control logic 606 generates the control signal 744 and outputs the control signal 744 to the flow control logic 726; hence, Pekkala et al. does not disclose or suggest rendering unavailable the system memory resource for the prescribed application resource in response to ***reception of*** the data flow interruption request.

Further, Pekkala et al. does not render unavailable the system memory resource for received packets, since there is no disclosure in Pekkala that any received packet from a link partner is dropped (due to the buffer memories being "rendered unavailable"); to the contrary, Pekkala et al. discloses that if the shared buffer 604 is unavailable due to consumption of free buffer space by received packets awaiting retransmission,[3] then inline spill buffers 612 are used to **hold data packets** received while the shared buffers 604 are full: (see steps 1208 and 1212 of Fig. 7, para. 107, 112:

> [0112] If the buffer control logic 606 determines during step 1208 that a shared buffer 604 is not available, the buffer control logic 606 asserts control signal 742 to ***cause the packet 300 from the receiver 722 to begin spilling into the inline spill buffer 612*** of FIG. 7 rather than flowing through the inline spill buffer 612, in step 1212.
>
> (Para. 112, lines 1-6).

Hence, Pekkala et al. does not disclose or suggest a memory controller that ***renders unavailable the system memory resources for the prescribed application resource in response to reception of the data flow interruption request***, because no packets are dropped; rather, Pekkala et al. stores any received packets in inline spill buffers 612 in response to detecting a depletion of the shared buffers 604. Consequently, Pekkala et al. does not render unavailable the system memory, but detects the unavailability of the system memory when the shared buffer is full.

For these and other reasons, the §102 rejection must be withdrawn.

Conclusion

A rejection for anticipation requires that the four corners of a single prior art document describe every element of the claimed invention, either expressly or inherently, such that a person of ordinary skill in the art could practice the invention without undue experimentation. *See Atlas Powder Co. v. Ireco Inc.*, 190 F.3d 1342, 1347, 51 USPQ2d 1943, 1947 (Fed. Cir. 1999); *In re Paulsen*, 30 F.3d 1475, 1478-79, 31 USPQ2d 1671, 1673 (Fed. Cir. 1994). Further, anticipation requires that Pekkala et al. discloses each and every element *as arranged in the claims under review*. As demonstrated above, the numerous inconsistencies by the Examiner demonstrate that Pekkala et al. fails to disclose each and every claim limitation as arranged in the claims.

[3]See, e.g., para. 12, 20-22, 76-79 of Pekkala et al.

For these and other reasons, the Examiner's rejection under §102 should be reversed.

For the reasons set forth above, it is clear that Appellant's claims 1-13 are patentable over the reference applied. Accordingly the appealed claims 1-13 should be deemed patentable over the applied reference. It is respectfully requested that this appeal be granted and that the Examiner's rejections be reversed.

Respectfully submitted,

Manelli Denison & Selter, PLLC

[illegible]

Leon R. Turkevich
Registration No. 34,035

Customer No. 20736
September 15, 2006